Los Alamos
NATIONAL LABORATORY
EST. 1943

Acquisition Services Management (ASM) Division
Subcontracts, ASM-SUB
P.O. Box 1663, Mail Stop D447
Los Alamos, New Mexico 87545
505-665-3814 / Fax 505-665-9022
E-mail: dknox@lanl.gov

DATE: August 29, 2013

Subject: **Question and Answer Set 7**
Trinity and NERSC-8 Computing Platforms Project
LA-UR-13-26826

Greetings:

Interested parties are advised of the following questions or concerns that have been submitted to the Trinity and NERSC-8 Project team and to the accompanying Project responses below:

Question/Issue 1

The first paragraph under the How To Run section of the IOR web page states *"blockSize will always be equal to transferSize"*. However, most of the tests specified in the N8TrinityScript IOR input file do not have blockSize=transferSize. Can we assume that each time transferSize is set that blockSize should be set to the same value?

Project Response 1

blockSize is the amount of data that each process will write/read contiguously in a segment. transferSize tells how much of that data will be written/read on each I/O. The N8TrinityScript is holding the amount of data per process segment (e.g. blockSize) constant and changing the transferSize so that in some tests the blockSize is written/read in more than one I/O. That is for the 10K tests, there 100 transfers. For the 100K test there are 10 transfers. For the 1m test there is 1 transfer.

Question/Issue 2

If the Offeror assumes the target I/O performance and JMTTI values are correct, then running the single IOR benchmark may be a problem. If this assumption is correct and it is not the intention of ACES and NERSC to run this benchmark for multiple days, can the NERSC-8/Trinity teams post an amendment or clarification to the benchmark?

[Offeror Comment] The elapsed time to run the single IOR benchmark using the N8TrinityScript input file appears to be longer than the expected JMTTI of the Trinity system. The script will write 1.5x system memory 24 times [(3 transfer sizes)*(4 types of test)*(2 repetitions)] and it also reads the same amount of data for a total data transfer of over 100 PB. Assuming the PFS is capable of writing or reading 80% of system memory in 20 minutes under all 24 requested scenarios, the optimal time to complete the run would be over 30 hours. However, some of the scenarios will likely run much slower than the optimal write case so the total run time of this one job would likely be several days.

Project Response 2

The NERSC-8/Trinity Project teams believe the confusion is based on the following excerpt from the IOR_RunRules-N8Trinity.docx document. The intention is to do test9 "a", "b" and "c" on a smaller prototype system that an Offeror has now. Then, "d" is not run, but is estimated for the full system that will be delivered based on the actual results from the prototype system. That is 1.5x the smaller prototype

system memory, not 1.5x the memory of the expected final delivery system. The answer to "d", the performance of the expected delivery system, is to be extrapolated based on the actual results to "a" - "c". Offerors should explain the math used to take the actual measurements on the smaller prototype system that led them to the predicted answer to "d".

Each of the tests previously listed will be run for the following processor/node count conditions:

a) The number of processors on one node that yields the peak results for a single node.

b) Process count for each node equals the number of cores on a node. Find the number of nodes that yields the peak results for the test system.

c) Process count for each node equals the number of cores on a node. Run on the total number of nodes that exist on the test system.

d) Process count for each node equals the number of cores on a node. Estimate the bandwidth when the test is run on all nodes that will exist on the delivered system (a projection based on data gathered).

Question/Issue 3

When reporting results for the MPI+X Optimized version of a benchmark, is it acceptable to use a modified version that uses single precision for part of the calculation as long as it's still passing verification?

Project Response 3

Please see the Benchmarking Run Rules for code optimization guidance. In particular the Benchmarking Run Rules note:

> Aggressive code changes that enhance performance are also permitted as long as the full capabilities of the code are maintained, the code can still pass validation tests, and the underlying purpose of the benchmark is not compromised.

Changes to the source code may be made so long as the following conditions are met:

- The rationale for and relative effect on performance of any optimization is described;
- Algorithms fundamental to the program are not replaced (since replacing algorithms may result in violations of correctness or program requirements or other chosen software decisions);
- All simulation parameters such as grid size, number of particles, etc., must not be changed;
- The optimized code execution must still result in correct numerical results;
- Any code optimizations must be made available to the general user community, either through a system library or a well-documented explanation of code improvements;
- Any library routine used must currently exist in an Offeror's supported set of general or scientific libraries, or must be in such a set when the system is delivered, and must not specialize or limit the applicability of the benchmark nor violate the measurement goals of the particular benchmark;
- Source preprocessors, execution profile feedback optimizers, etc. are allowed as long as they are, or will be, available and supported as part of the compilation environment for the delivered systems;
- Only publicly available and documented compiler switches shall be used;
- Finally, the same code optimizations must be made for all runs of a benchmark. For example, one set of code optimizations may not be made for the "small" case while a different set of optimizations made for the "large" case.

Any specific code changes and the runtime configuration used must be clearly documented with a complete audit trail and all supporting documentation included in the submission. Trinity/NERSC-8 will be the final judge of whether optimizations will be acceptable.

Question/Issue 4

From the Technical Requirements document, section 3.5.1, in the customer spreadsheet of results, the OMB tab of the spreadsheet – What is the definition of Concurrency for osu_allgather, osu_allreduce, and osu-barrier – MPI ranks per server?

Project Response 4

The instructions require that the osu_allgather, osu_allreduce, and osu-barrier benchmarks be run with one MPI task per core over all nodes in the system. The spreadsheet "Concurrency Used" entries refer to the total number of MPI ranks that were used to run the benchmarks on a reference system (cell E125) or that would be used on a proposed system (cell K125).

Darren Knox

Acquisition Services Management
Los Alamos National Security, LLC
Los Alamos National Laboratory